

Requirement Document

OTT PLATFORM Management

Objective:

- **The main objective of this project is to allow or restrict different type of users based on the subscription to access the content in OTT Platform .**
- **Only admin can add or delete the content from platform.**
- **Displays the content by language, genre, cast, region, rating.**

Scope:

The main scope of this project is to allow the user to access for the desired movies , tv_shows and series. This application is developed in such a way that based on the membership user can access the Content . It will display all details of content like cast, language and genre. Only Admin can add the content in table. A new user can signup and the details are added in user table.

Functional Requirements :

- User Store User Details
- Get Password and Logintype of login id for Validation
- Get User Details
- Update Login Time of User
- View Membership Validity
- Upgrade Membership
- View The Contents
- View Cast of Selected Content

- View Language of Selected Content
- View Genre of Selected Content
- View User Name and Login Time for Admin
- Admin Adds New Content

Application Functionality:

3.1 Signup

User Enters login_id , password, firstname, lastname, dob, mobile no., email-id, country and after validation. The Details are entered in the login and user table.

3.2 Login In

User Enters login_id and password. Based on the login_id function gets password and logintype from database and validates the password.

3.2.1 Update LoginTime

Update login time of the logged in user in the database

3.2.2 IF User

3.2.2.1 View Profile

Function gets Name, DOB, Mobile-NO, Email-Id, Subscription from database

3.2.2.2 View Validity

Function gets validity of membership

3.2.2.3 Upgrade Membership

User can upgrade their membership , if user doesn't have membership he can upgrade to VIP or PREMIUM. If user has VIP he can upgrade to PREMIUM

3.2.2.4 View All Contents

User can view all Contents from Data base

3.2.2.4.1 View Cast of Content

User can view cast and role name of content from database

3.2.2.4.2 View Language of Content

User can view languages of Content from database

3.2.2.4.3 View Genre of Content

User can view Genre of Content from database

3.2.3 IF ADMIN

3.2.3.1 View User log

Admin can view user_name and login time of all users.

3.2.3.2 Add Content

Admin enters Content_Name, Content_Type, Release Date, Rating, Censor and Access Type and adds into the database

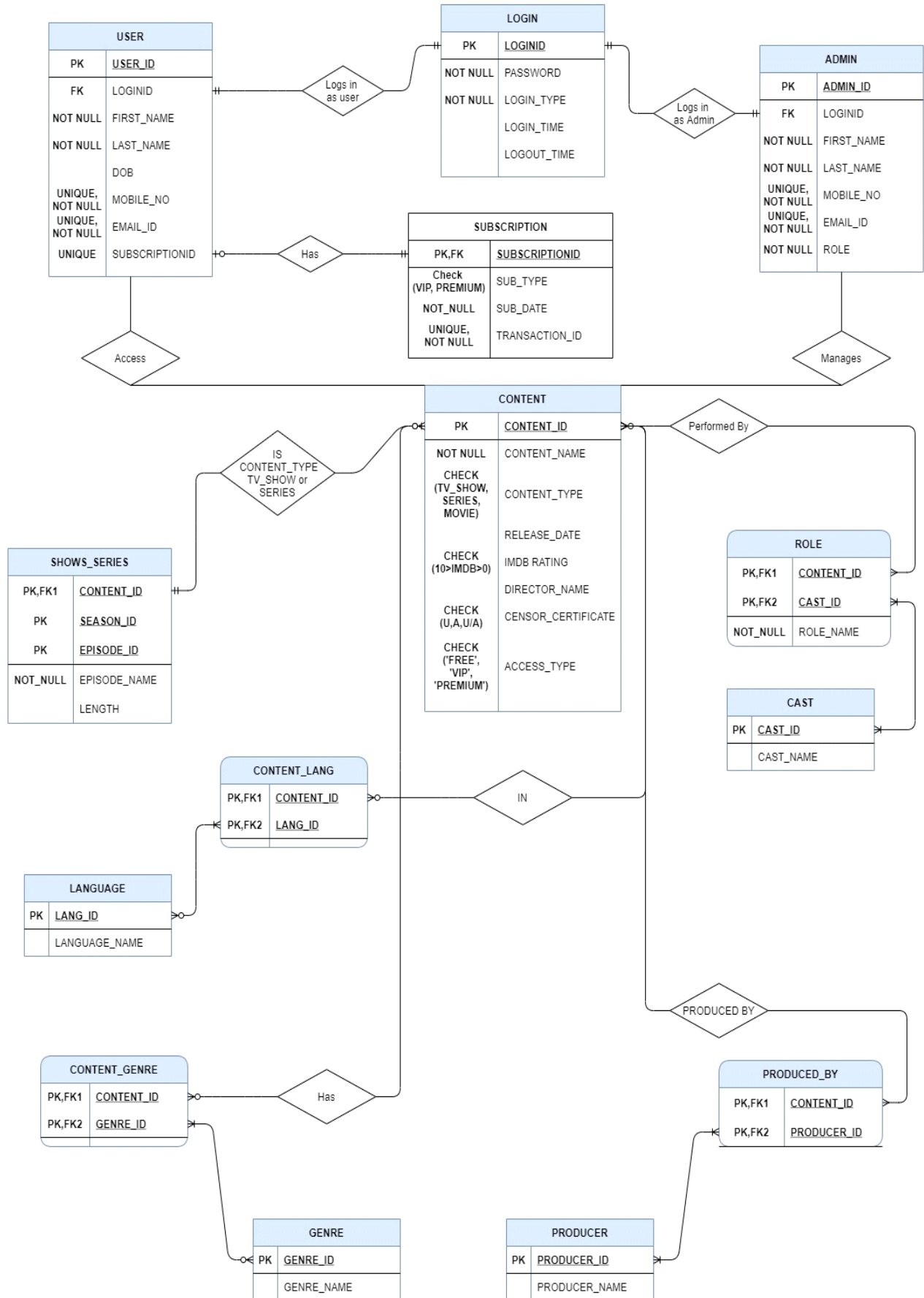
Application User:

The Application can be accessed by User to view content.

The Application can be used by Admin to Access and Manage the database

Back end: Oracle DatabaseER

Diagram:



Database Design:

a. Table Structure for LOGIN table

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	LOGIN_ID	VARCHAR2(20 BYTE)	No	(null)	1 (null)	
2	PASSWORD	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3	LOGIN_TIME	TIMESTAMP(6)	Yes	(null)	3 (null)	
4	LOGOUT_TIME	TIMESTAMP(6) WITH TIME ZONE	Yes	(null)	4 (null)	
5	LOGIN_TYPE	VARCHAR2(10 BYTE)	Yes	(null)	5 (null)	

b. Table Structure for USERS table

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	USER_ID	VARCHAR2(8 BYTE)	No	(null)	1 (null)	
2	LOGIN_ID	VARCHAR2(20 BYTE)	Yes	(null)	2 (null)	
3	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4	LAST_NAME	VARCHAR2(20 BYTE)	No	(null)	4 (null)	
5	DOB	DATE	Yes	(null)	5 (null)	
6	MOBILE_NO	NUMBER(10,0)	No	(null)	6 (null)	
7	EMAIL_ID	VARCHAR2(35 BYTE)	No	(null)	7 (null)	
8	SUBSCRIPTION_ID	VARCHAR2(10 BYTE)	Yes	(null)	8 (null)	
9	COUNTRY	VARCHAR2(20 BYTE)	No	(null)	9 (null)	

c. Table Structure for ADMIN table

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ADMIN_ID	VARCHAR2(20 BYTE)	No	(null)	1 (null)	
2	FIRST_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3	LAST_NAME	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4	MOBILE_NO	NUMBER(10,0)	No	(null)	4 (null)	
5	EMAIL_ID	VARCHAR2(35 BYTE)	No	(null)	5 (null)	
6	ROLE	VARCHAR2(20 BYTE)	No	(null)	6 (null)	

d. Table Structure for CONTENT_TABLE

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 CONTENT_NAME	VARCHAR2(35 BYTE)	No	(null)	2 (null)	
3 CONTENT_TYPE	VARCHAR2(10 BYTE)	Yes	(null)	3 (null)	
4 RELEASE_DATE	DATE	Yes	(null)	4 (null)	
5 IMDB_RATING	NUMBER(2,1)	Yes	(null)	5 (null)	
6 DIRECTOR_NAME	VARCHAR2(20 BYTE)	Yes	(null)	6 (null)	
7 CENSOR_CERTIFICATE	CHAR(3 BYTE)	Yes	(null)	7 (null)	
8 ACCESS_TYPE	VARCHAR2(10 BYTE)	Yes	(null)	8 (null)	

e. Table Structure for CAST table

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 CAST_ID	NUMBER(6,0)	No	(null)	1 (null)	
2 CAST_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

f. Table Structure for ROLE_TABLE

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 CAST_ID	NUMBER(6,0)	No	(null)	2 (null)	
3 ROLE_NAME	VARCHAR2(20 BYTE)	No	(null)	3 (null)	

g. Table Structure for GENRE table

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 GENRE_ID	NUMBER(3,0)	No	(null)	1 (null)	
2 GENRE_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

h. Table Structure for CONTENT_GENRE table

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 GENRE_ID	NUMBER(3,0)	No	(null)	2 (null)	

i. Table Structure for LANGUAGE table

↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1 LANGUAGE_ID	NUMBER(3,0)	No	(null)	1 (null)	
2 LANGUAGE_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

j. Table Structure for CONTENT_LANGUAGE table

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 LANGUAGE_ID	NUMBER(6,0)	No	(null)	2 (null)	

k. Table Structure for PRODUCERS table

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PRODUCER_ID	NUMBER(6,0)	No	(null)	1 (null)	
2 PRODUCER_NAME	VARCHAR2(20 BYTE)	No	(null)	2 (null)	

l. Table Structure for PRODUCERS_BY table

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 PRODUCER_ID	NUMBER(6,0)	No	(null)	2 (null)	

m. Table Structure for TV_SHOW/SERIES table

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 CONTENT_ID	VARCHAR2(6 BYTE)	No	(null)	1 (null)	
2 SEASON_NO	NUMBER(3,0)	No	(null)	2 (null)	
3 EPISODE_NO	NUMBER(5,0)	No	(null)	3 (null)	
4 EPISODE_NAME	VARCHAR2(30 BYTE)	No	(null)	4 (null)	
5 LENGTH	VARCHAR2(6 BYTE)	Yes	(null)	5 (null)	

n. Table Structure for SUBSCRIPTION table

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 SUBSCRIPTION_ID	VARCHAR2(10 BYTE)	No	(null)	1 (null)	
2 SUBSCRIPTION_TYPE	VARCHAR2(20 BYTE)	Yes	(null)	2 (null)	
3 SUBSCRIPTION_DATE	DATE	No	(null)	3 (null)	
4 TRANSACTION_ID	NUMBER(10,0)	No	(null)	4 (null)	

Constraint:

The 'LOGINID' in LOGIN table is primary named as LOGIN_LOGINID_pk. .

'PASSWORD' should not be null named as LOGIN_PASSWORD_NN and 'LOGIN_TYPE' named as LOGIN_LOGINTYPE_NN should not be null.

The 'USER_ID' in USERS table is primary named as USERS_USERID_pk.

The 'FIRST_NAME', 'LAST_NAME','COUNTRY' in USERS table is not null constraint.

'Mobile NO','Email_Id' in Users Table should be unique and not null

'SUBSCRIPTION_ID' should be of length 10.

In USERS table 'LOGIN_ID' of the USERS table is the child column and acts as foreign_key_constraints named as USERS_LOGINID_fk.

The 'ADMIN_ID' in ADMIN table is primary named as ADMIN_ADMINID_pk.

'ADMIN_ID' should start with 'A' with leading '0's of 6

The 'FIRST_NAME', 'LAST_NAME','COUNTRY' in ADMIN table is not null constraint.

'Mobile NO','Email_Id' in ADMIN Table should be unique and not null

In ADMIN table 'LOGIN_ID' of the ADMIN table is the child column and acts as foreign_key_constraints named as ADMIN_LOGINID_fk.

The 'SUBSCRIPTIONID' in SUBSCRIPTION table is primary named as SUBSCRIPTION_SUBSCRIPTIONID_pk.

The 'SUB_DATE' in SUBSCRIPTION table is not null constraint.

'TRANSACTION_ID' in Users Table should be unique

In USERS table 'SUBSCRIPTIONID' of the SUBSCRIPTION table is the child column and acts as foreign_key_constraints named as SUBSCRIPTION_SUBSCRIPTIONID_fk.

The 'CONTENT_ID' in CONTENT_TABLE is primary named as CONTENT_CONTENTID_pk.

'CONTENT_ID' should start with 'C' with leading '0's of 6

The 'CONTENT_NAME', 'CONTENT_TYPE' in CONTENT_TABLE table is not null constraint.

'CERTIFICATE','ACCESS_TYPE' in CONTENT_TABLE has check constraint 'certificate' should be in 'U','A','U/A','access_type' should be in 'VIP','PREMIUM'

The 'CONTENT_ID','SEASON_NO','EPISODE_NO in TV_SHOW/SERIES table is composite key named as SERIES_COMPOSITEKEY_pk.

The 'EPISODE_NAME' in TV_SHOWS/SERIES table is not null constraint.

The 'CONTENT_ID','CAST_ID' in ROLE table is primary named as ROLE_COMPOSITEKEY_pk.

The 'CAST_NAME' in TV_SHOWS/SERIES table is not null constraint.

The 'CAST_ID' in CAST table is primary named as CAST_CASTID_pk.
The 'CAST_NAME' in CAST table is not null constraint.

The 'LANGUAGE_ID' in LANGUAGE table is primary named as LANGUAGE_LANGID_pk.

The 'LANGUAGE_NAME' in LANGUAGE table is not null constraint.

The 'CONTENT_ID','LANGUAGE_ID' in CONTENT_LANG table is composite key named as CLANGUAGE_COMPOSITEKEY_pk.

The 'GENRE_ID' in GENRE table is primary named as GENRE_GENREID_pk.
The 'GENRE_NAME' in GENRE table is not null constraint.

The 'CONTENT_ID','GENRE_ID' in CONTENT_GENRE table is composite key named as CGENRE_COMPOSITEKEY_pk.

The 'PRODUCER_ID' in PRODUCER table is primary named as PRODUCER_PRODUCERID_pk.

The 'PRODUCER_NAME' in LANGUAGE table is not null constraint.

The 'CONTENT_ID','PRODUCER_ID' in PRODUCED_BY table is composite key named as PRODUCEDBY_COMPOSITEKEY_pk.

Functional Requirements Details :

1.Login Object:

- i) Add new user details.
- ii) Get login password and login type and validate.

iii) Update login time as soon as user logs in.

2. User Object:

- i) Shows User Details.
- ii) View Membership Validity.
- iii) Upgrade Membership.

3. Content Object:

- i) Shows all the Content to the users.
- ii) Shows Cast and Role Name of the Content
- iii) Shows Language of the Content.
- iv) Shows Genre of the Content.

4. Admin Object:

- i) Shows UserName and login time of user.
- ii) Admin adds new Content in DataBase.

Classification: Internal

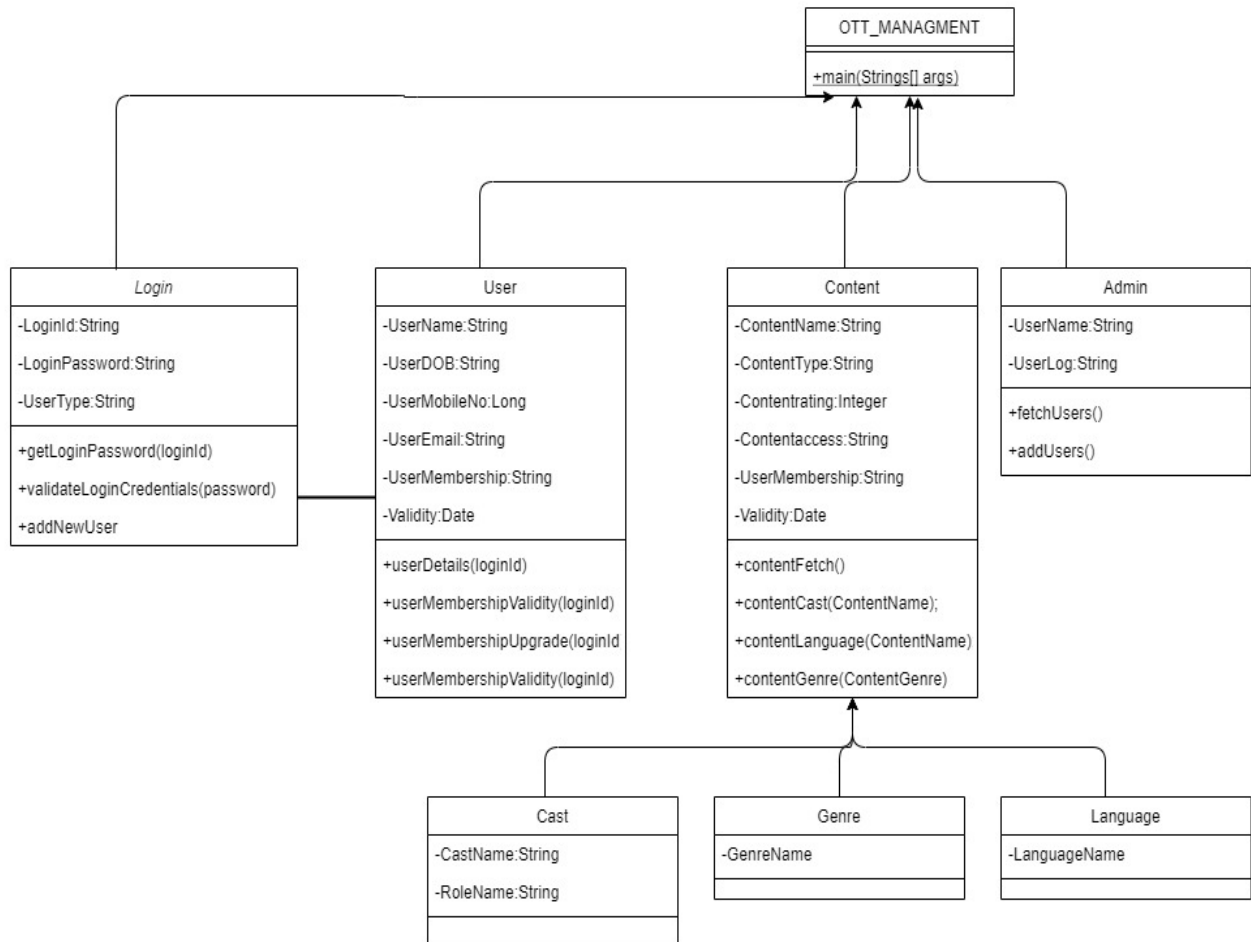
6. Login_service:

- i) The Login_service functionalities are applied to the Login, User before getting into the database. The User should be filtered in terms of loginid, password, mobilenumber, email_id. Login Credentials should be checked before entering into the application

7. Admin_service:

- i) The Admin_service functionalities are applied to the Admin before getting into the database. The Content details entered by Admin should be filtered in terms of contenttype, contentrating, contentcensor, contentaccess.

Class Diagram:



Front End: Java

Java Class Used:

Domain Object/ Value Object:

Purpose of class:

1. To store the Login details such as loginId, Password,LoginType

2. To store the User details such as UserName, UserDOB, UserMobileNo, UserEmailid, UserMembership, MembershipValidity.
3. To store the Admin details such as Userlog, UserName
4. To store the Content details such as ContentName, ContentType, ContentRating, ContentAccessType
5. To store the Cast details such as CastName, RoleName
6. To store the Language details such as Language Name
7. To store the Genre details such as Genre Name

Package	Class Name	methods
com.birlasoft.ottplatform.model	AdminVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	CastVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	ContentVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	GenreVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	LanguageVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	LoginVO	Constructor, Getters and Setters, toString
com.birlasoft.ottplatform.model	UserVO	Constructor, Getters and Setters, toString

Service Layer:

Purpose of class:

1. To validate login name, password
2. To validate loginid, password, mobilenno, emailid such that id should not be less than 8 , password should have uppercase, lowercase, number, (%*%), email should be follow pattern @.com
3. To validate content type, content rating, content censor, content access such that content type is either 'MOVIES', 'SERIES', 'TV_SHOWS' and content rating should be between 0-10, content censor in 'U', 'A' , 'U/A' , content access in 'VIP', 'PREMIUM', 'FREE'

Package	Class Name	Methods
com.birlasoft.ottplatform.service	LoginService	validateLogin
com.birlasoft.ottplatform.service	LoginService	validatePassword
com.birlasoft.ottplatform.service	LoginService	validateloginid
com.birlasoft.ottplatform.service	LoginService	validatemobilenno
com.birlasoft.ottplatform.service	LoginService	validateemailid
com.birlasoft.ottplatform.service	AdminService	contentTypeValidate
com.birlasoft.ottplatform.service	AdminService	contentRatingValidate
com.birlasoft.ottplatform.service	AdminService	contentCensorValidate
com.birlasoft.ottplatform.service	AdminService	contentAccessTypeValidate

Data Access Layer:

Purpose of class: Access and add the data to the database

Package	Class Name	Methods
com.birlasoft.ottplatform.dao	AdminDAO	Public List<Admin> fetchUsers(); Public void addContent(String cname, String ctype, String crelease, float crating, String ccensor, String caccess)
com.birlasoft.ottplatform.dao	ContentDAO	Public List<Content> fetchContent(); Public List<Cast> contentCast(String contentName) Public List<Language> contentLanguage(String contentName)

		Public List<Genre> contentGenre(String contentName)
com.birlasoft.ottplatform.dao	LoginDAO	Public void getLoginPassword(String loginID) Public void updateLoginTime(String loginID) Public void addNewUser(String loginid, String password, String firstname, String lastname, String DOB, Long mobile, String email id, String Country)
com.birlasoft.ottplatform.dao	UserDAO	Public void userDetails(String loginId) Public void userMembershipValidity(String loginId) Public void userMembershipUpgrade(String loginId, String membership)

Utility:

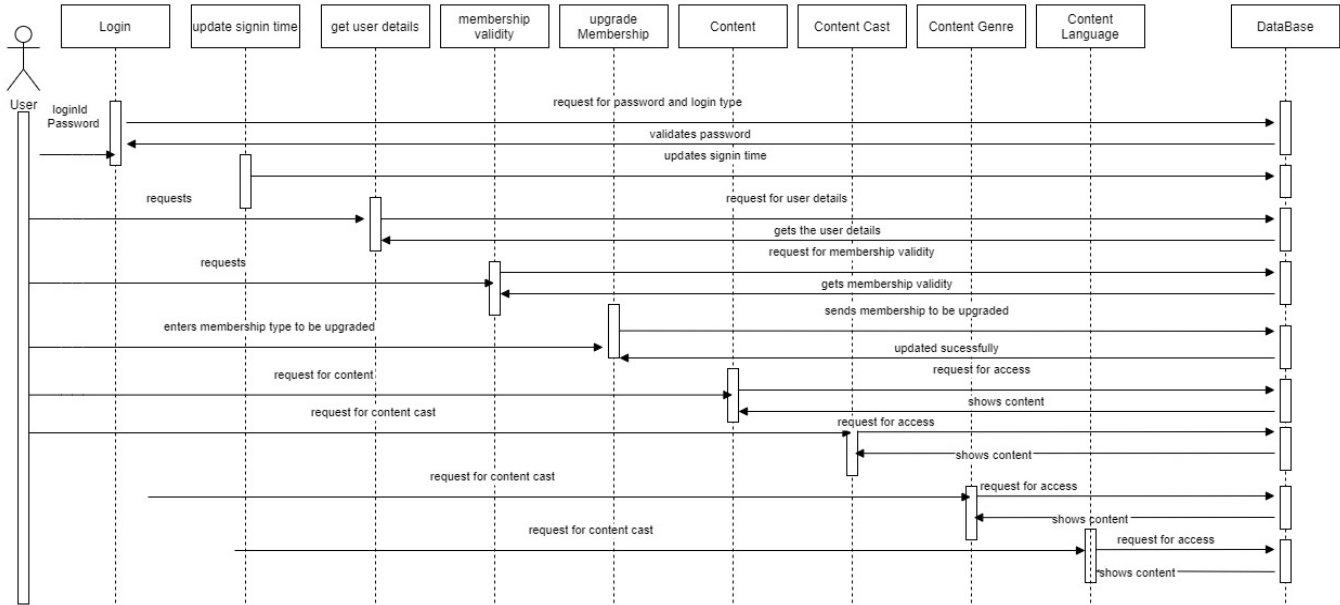
Purpose of class:

For Database Connectivity

Package	ClassName	Methods
com.birlasoft.ottplatform.util	DBOracleConnect	Private static void loadDrivers()
com.birlasoft.ottplatform.util	DBOracleConnect	Public static

		Connection getOracleConnection()
com.birlasoft.ottplatform.util	DBOracleConnect	Public static void closeOracleConnection

Sequence Diagram:



PLSQL package description:

- **PKG_ADDNEWUSER** : inserts in login and users table.

Package Name	Procedure Name	Input Parameters	Output
PKG_ADDNEWUSER	PRC_NEWUSER	P_loginid, P_Password, P_firstname, P_lastname, P_DOB, P_mobile_no, P_emailid,	

		P_country	
--	--	-----------	--

- **FETCH_LOGINDETAILS**

Package Name	Procedure Name	Input Parameters	Output
FETCH_LOGINDETAILS	LOGIN_PRC	P_loginid	P_password, P_login_type

- **UPDATE_LOGINDETAILS**

Package Name	Procedure Name	Input Parameters	Output
UPDATE_LOGINDETAILS	SIGNIN_PRC	P_loginid	

- **PKG_GETUSERDETAILS**

Package Name	Procedure Name	Input Parameters	Output
PKG_USERDETAILS	PRC_GETUSERDETAILS	P_loginid	P_Name, P_DOB, P_MobileNO , P_EmailId, P_subscription

- **PKG_MEMBERSHIP_VALIDITY**

Package Name	Procedure Name	Input Parameters	Output
PKG_MEMBERSHIP_VALIDITY	PRC_	P_loginid	P_Validity

	VALIDITY		
--	----------	--	--

- **PKG_MEMBERSHIP_UPGRADE**

Package Name	Procedure Name	Input Parameters	Output
PKG_MEMBERSHIP_UPGRADE	PRC_UPGRADE	P_loginid, P_MEMBERSHIP	

- **PKG_FETCHCONTENT**

Package Name	Procedure Name	Input Parameters	Output
PKG_FETCHCONTENT	PRC_CONTENT		rc_cursor (ref cursor)

- **PKG_FETCHCAST**

Package Name	Procedure Name	Input Parameters	Output
PKG_FETCHCAST	PRC_CAST		rc_cursor (ref cursor)

- **PKG_FETCHLANGUAGE**

Package Name	Procedure Name	Input Parameters	Output
PKG_FETCHLANGUAGE	PRC_LANGUAGE		rc_cursor (ref cursor)

- **PKG_FETCHGENRE**

Package Name	Procedure Name	Input Parameters	Output
PKG_FETCHGENRE	PRC_GENRE		rc_cursor (ref cursor)

- **PKG_FETCHUSERS**

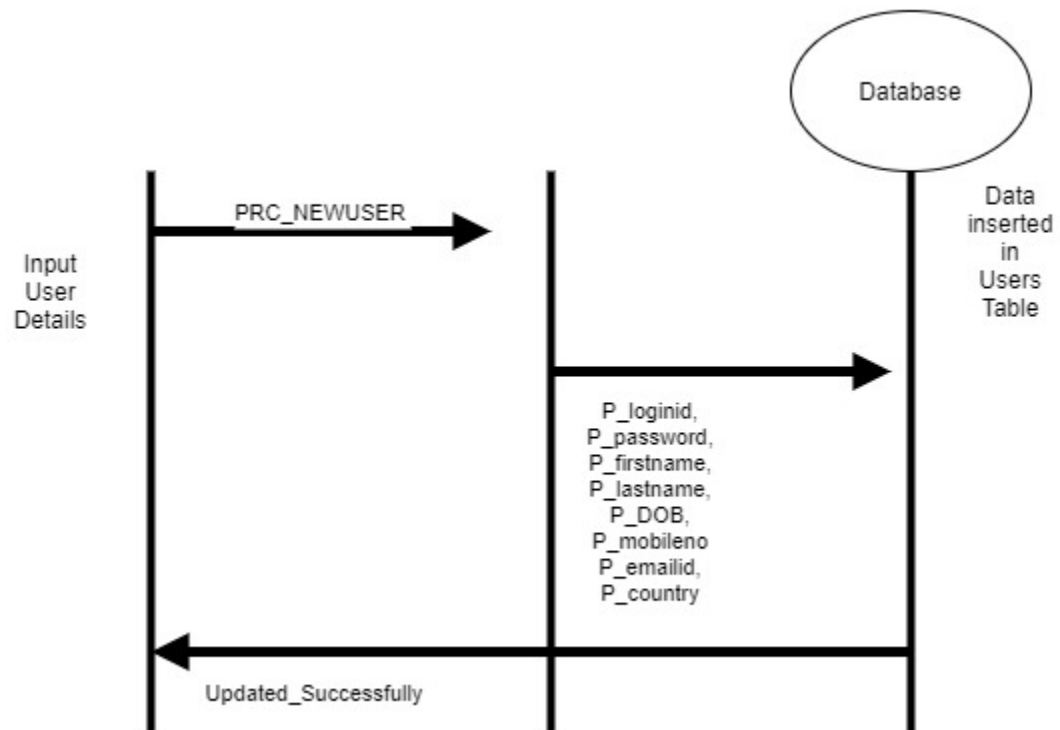
Package Name	Procedure Name	Input Parameters	Output
PKG_FETCHCONTENT	PRC_USERSLOG		rc_cursor (ref cursor)

- **PKG_ADDNEWCONTENT**

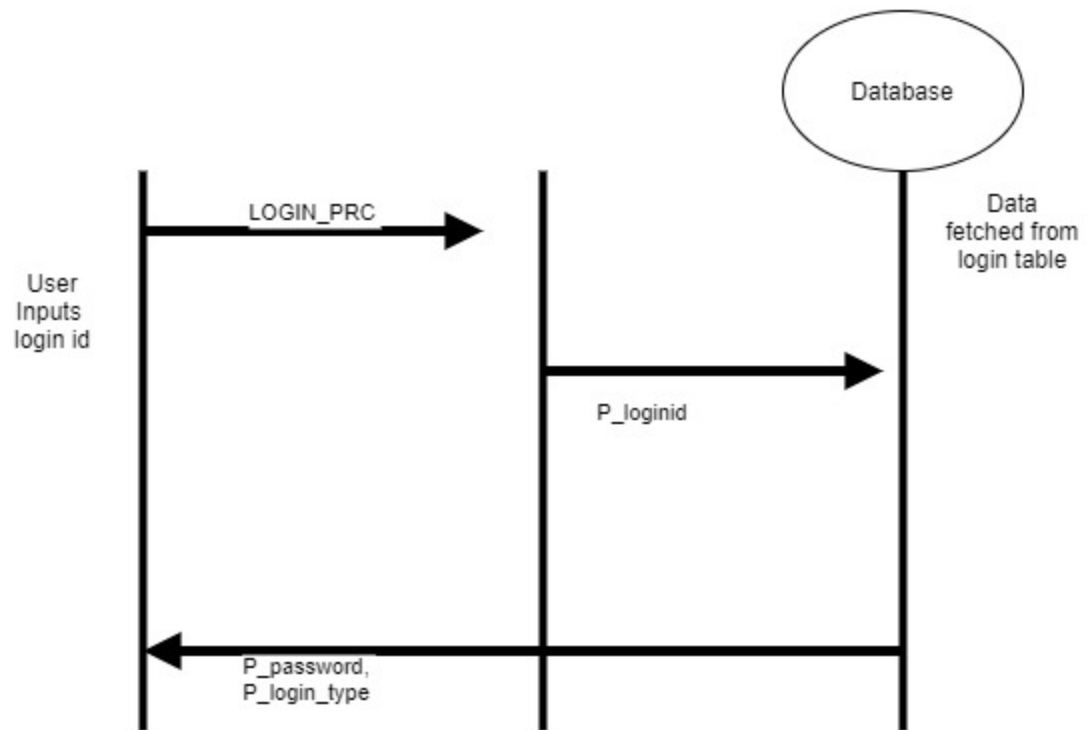
Package Name	Procedure Name	Input Parameters	Output
PKG_ADDNEWCONTENT	PRC_NEWCONTENT	P_contentname, P_contenttype, P_releasedate, P_rating, P_censor, P_access	

Data Flow Diagram Per Module:

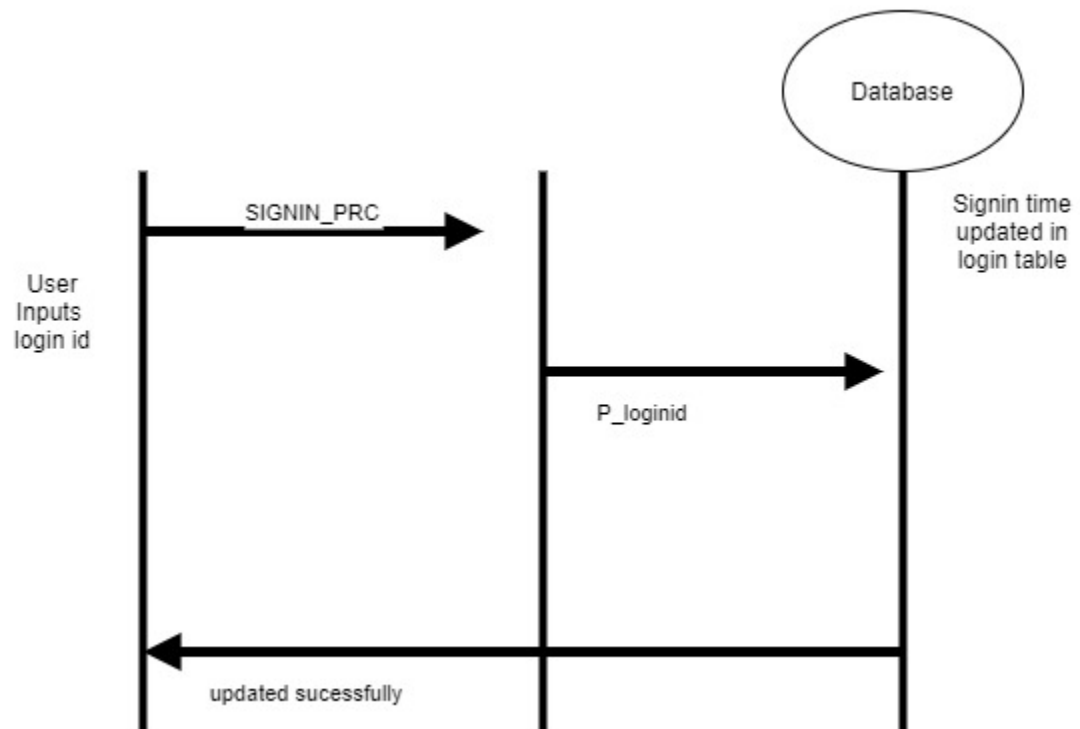
1. Signup inserts in login and users table.



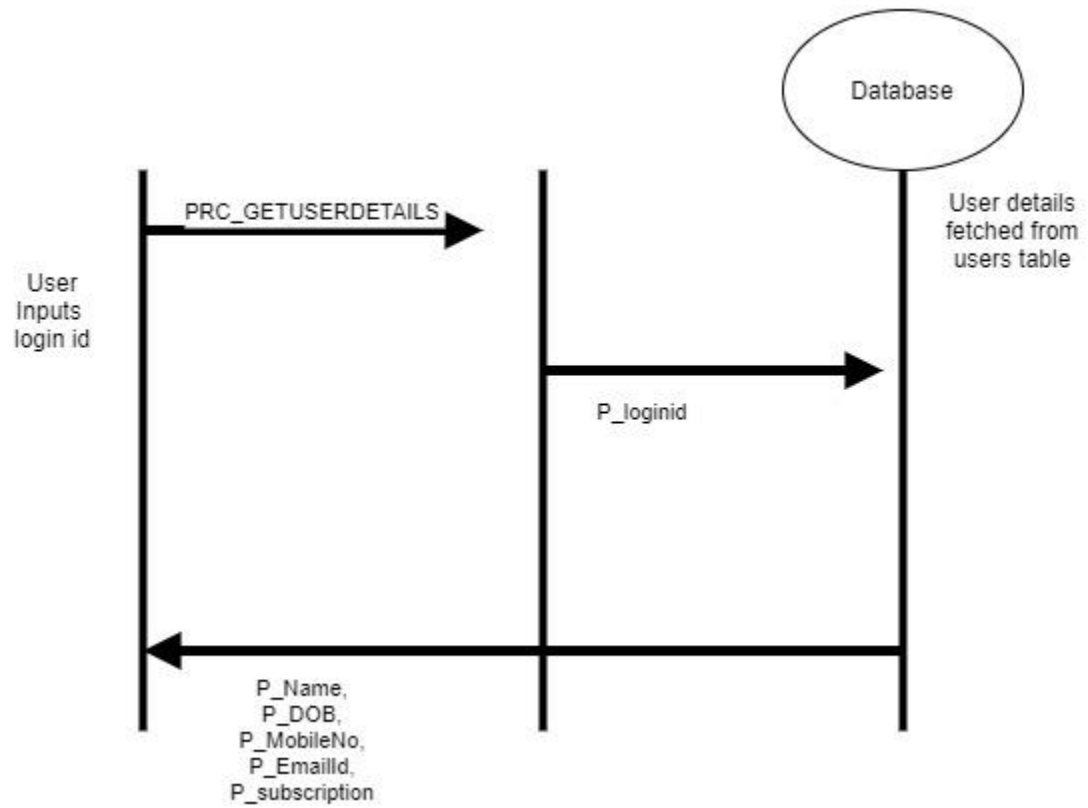
2. Get LoginType and Password of Users.



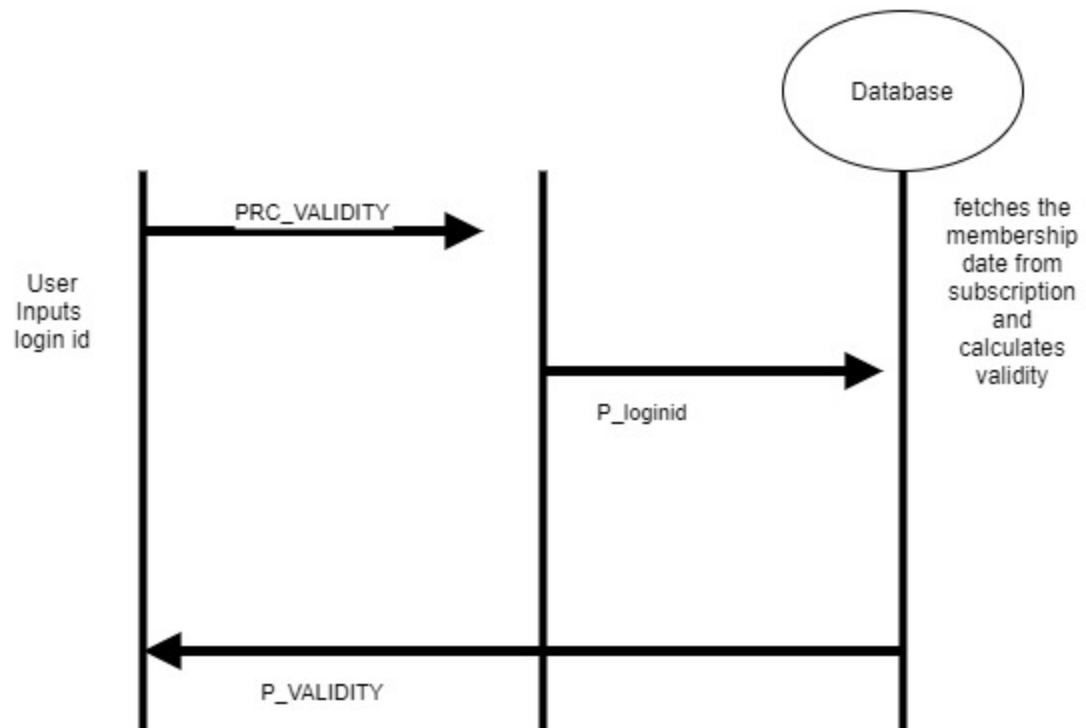
3. Updating Signin time in login table



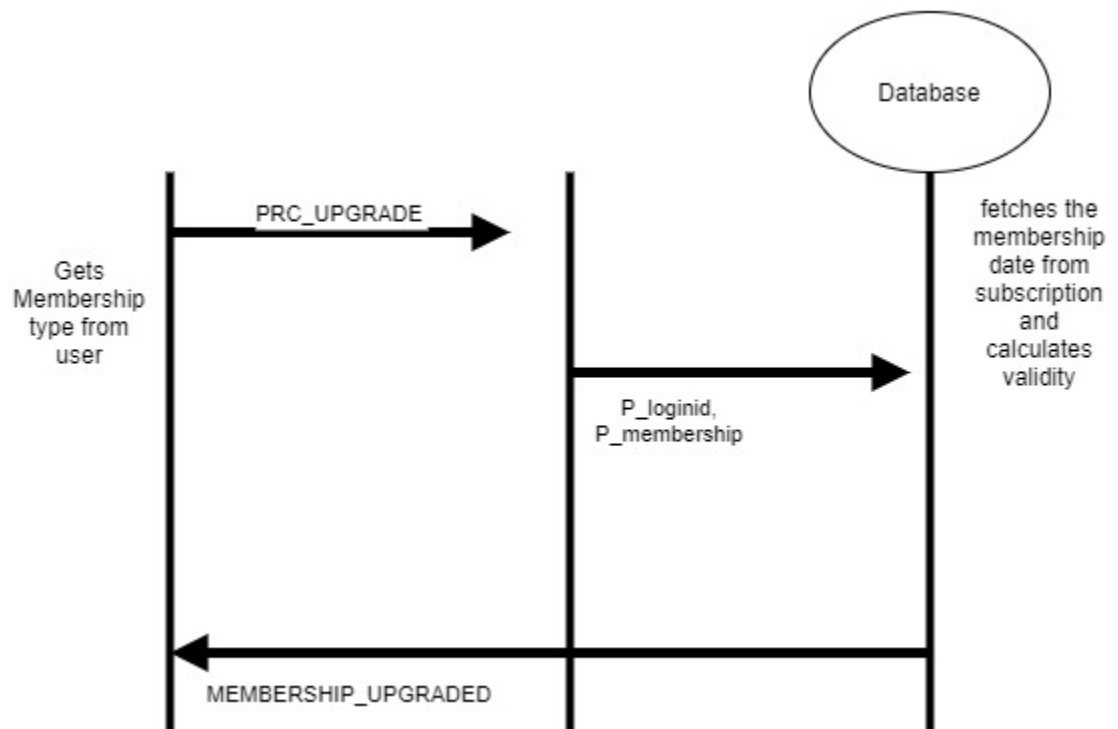
4. Get User Details from Users table



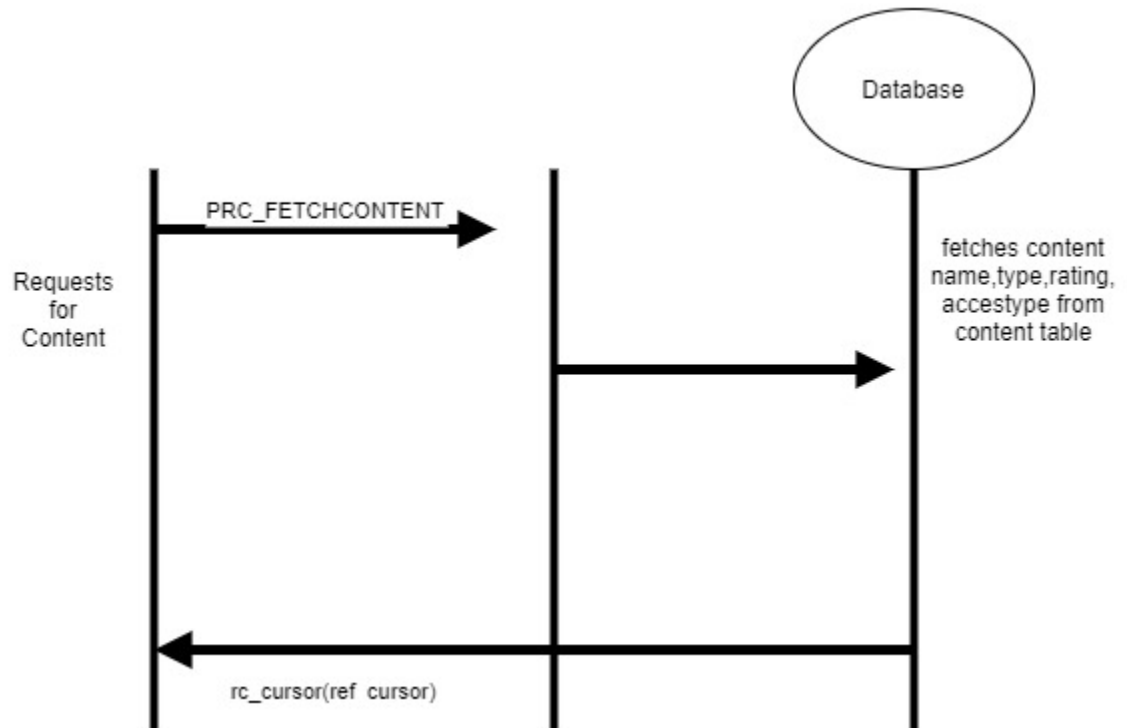
5. Get Membership Validity



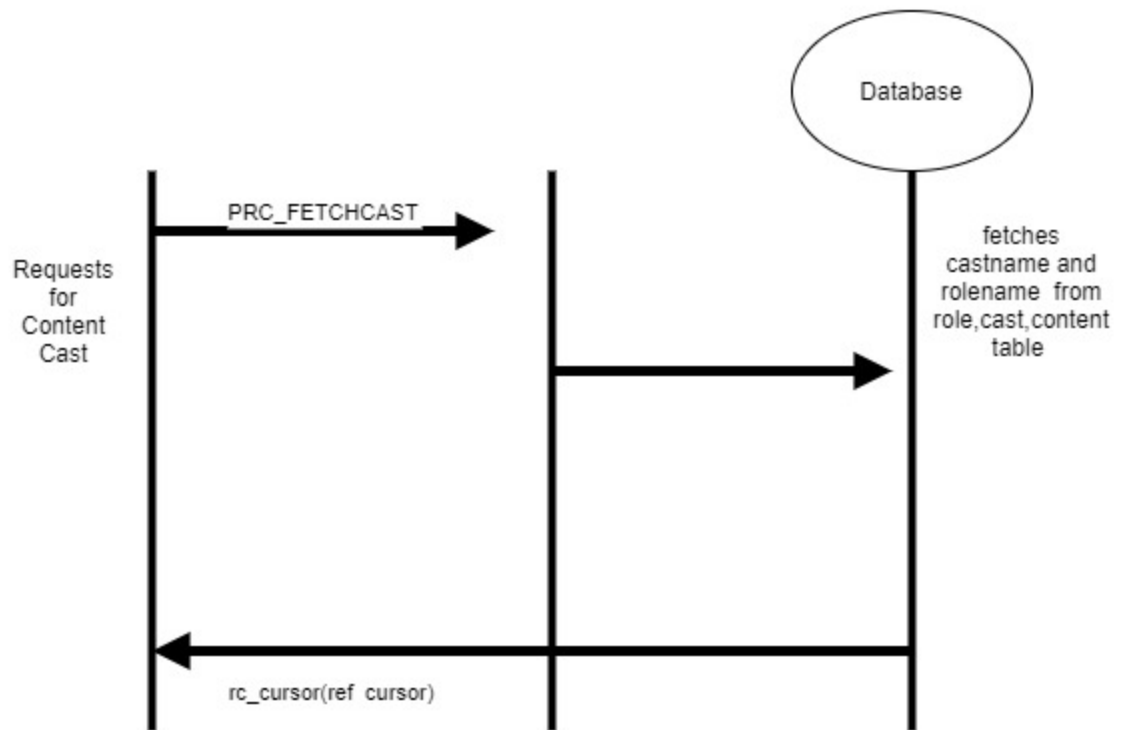
6. Upgrade Membership



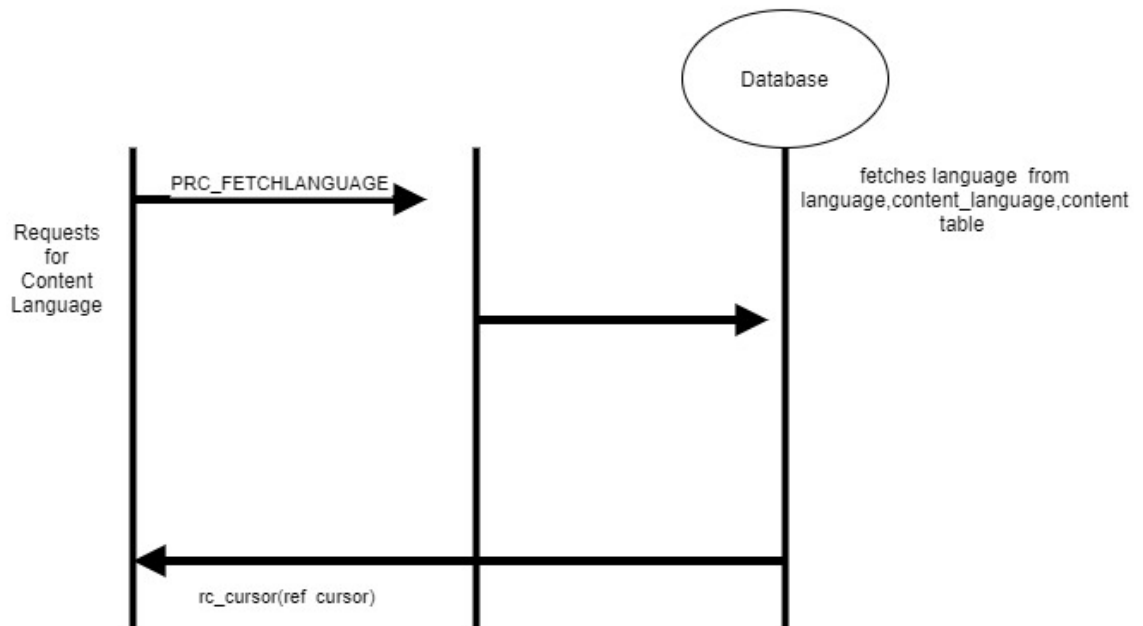
7. Get Content Details from Content Table



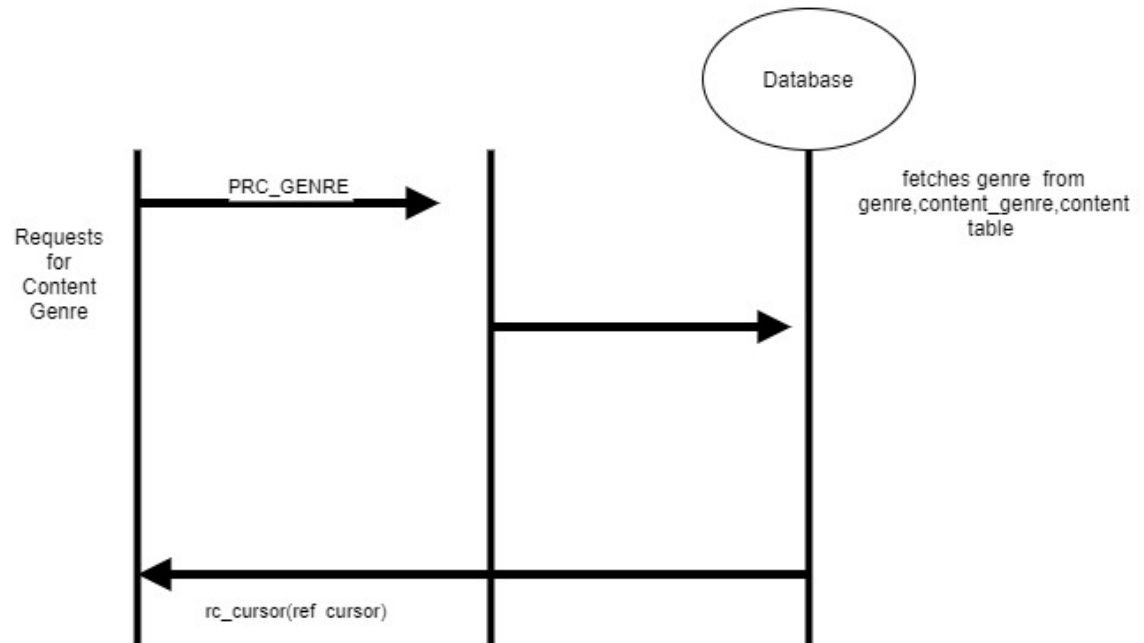
8. Get Cast and Role Details of Content



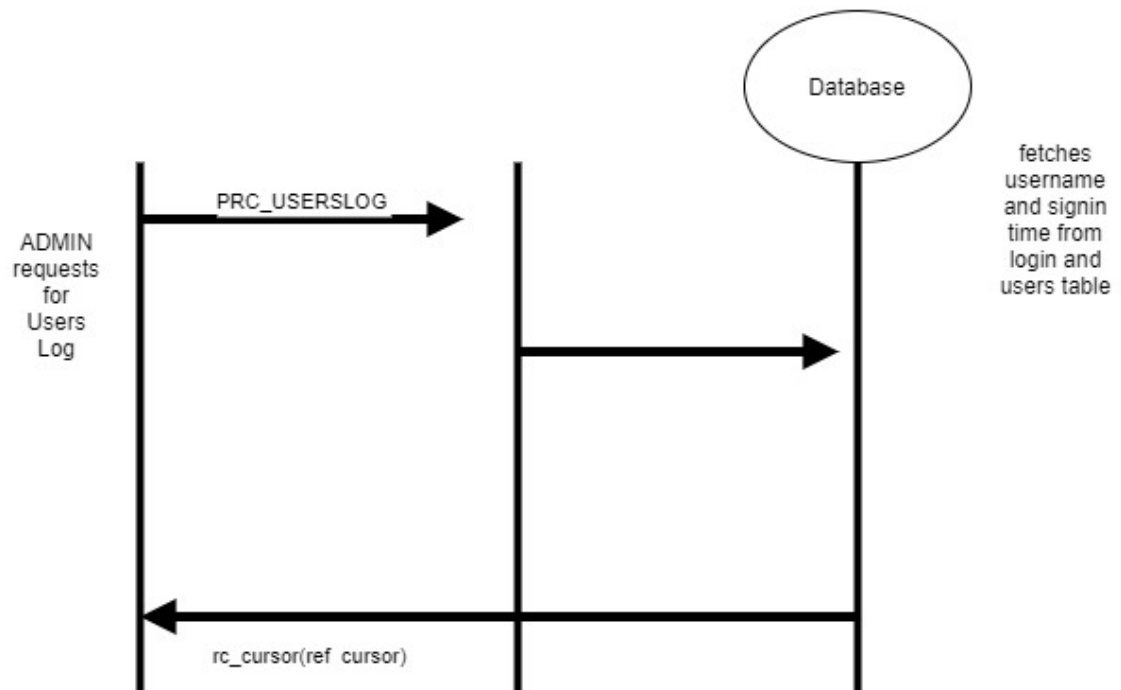
9. Get Language of Content



10. Get Genre of Content



11. Get Users Log



12.Add New Content into Content Table

